CoolFish:一个兼容 BitTorrent 的 P2P 视频点播、直播系统

李哲中 张铁赢 刘悦 程学旗

摘要:随着网络在线视频的广泛应用，对等传输（P2P）技术越来越受到业界的重视。我们开发的 CoolFish 是一个基于对等传输技术，集视频点播、直播于一体的流媒体系统。在本文中，我们基于 CoolFish 系统，对目前流行的视频传输技术做了较为详细的探讨和阐述，并对 CoolFish 的架构、功能和模块设计进行了全面介绍，另外，我们对 CoolFish 系统中涉及到的对等传输关键技术和算法进行了深入探讨。

关键词: CoolFish，网络视频，流媒体，对等传输（P2P），直播，点播，BitTorrent

1 引言

近年来，网络视频服务（流媒体）是最流行的互联网应用之一。所谓网络视频，是指用户利用互联网在线观看视频节目。例如，不用电视机，用户可以直接在网络上观看电视直播；再如，不用下载整个文件，用户便可在线点播喜欢的电影。如此方便的应用，自然得到了数以亿计的互联网用户的极大欢迎。网络视频网站更是不断涌现，如 YouTube[1]，tudou[2]，PPLive[3]，UUSee[4]，PPStream[5]等。

实际上，流媒体能够得到大规模的应用，是以互联网硬件技术的迅猛发展，尤其是终端用户接入带宽的快速提高为基础的。在数年前，家庭终端用户的互联网应用还仅仅局限于浏览网页，而现在，随着“最后一公里”瓶颈[1]的极大缓解，越来越多的用户选择在线观看视频节目，这也使互联网应用变得更加丰富多彩。然而，最后一公里的解决导致了第一公里问题[2]日渐突出。用户庞大的数据下载导致媒体内容服务器常常不能满足需求。例如，目前最大的视频分享服务提供商 YouTube，每个月要为提供的数据传输支付上百万美金[6]。不仅如此，YouTtube 还要装备具有 400 个节点 10G 高速网络的超级服务器来提供数据传输服务。为了保证如此庞大的下载需求，工业界和学术界纷纷提出了相应的视频传输技术。比较流行的有：增强的客户/服务器模式，内容分发网络（Content Delivery Network，CDN），对等传输（P2P，peer to peer）以及多种方式相结合的架构。在本文中，我们将对上述方法进行较为详细的阐述和对比。同时，我们设计并实现部署了一个对等传输流媒体系统——CoolFish。CoolFish 能够兼容 BitTorrent 网络并提供视频直播和点播服务。本文对 CoolFish 的架构、功能和模块设计进行了全面介绍，并且对 CoolFish 系统中涉及到的对等传输关键技术和算法进行了深入探讨。

2 网络视频传输技术

2.1 视频传输方式

目前，视频服务提供者用来传输视频的方式可分为以下两类：(1).下载：接收者所获得

---

[1] 最后一公里问题是指终端用户接入带宽与骨干网相比过低，导致互联网服务的用户体验很差，运营商主干带宽不能充分利用。

[2] 第一公里问题是指服务器端上传带宽不能满足用户需求。

的视频内容以文件形式储存在硬盘之上，下载完成后，接收者方可观看；(2).流媒体：接收者以在线观看的形式接收视频内容。在本节内容中，我们将详细讨论上述视频传输方式。

2.1.1 下载

下载方式是指接收者所获得的视频内容以文件的形式储存在用户的硬盘之上。，这种方式可以使用集中式文件服务器或者对等传输网络进行内容分发。像苹果公司（Apple）的 I-Tunes 视频以及提供好莱坞电影下载的 Cinemanow 都是这一类基于下载的网络视频服务。在视频内容被下载之后，文件可以被传送至移动视频设备之上，或者在对等传输网络之中合法地共享。

2.2 流媒体

流媒体则为用户提供了一种边下载边播放的方式。在此方式下，视频内容被缓存在本地。传统的流媒体技术需要集中式流媒体服务器来分发视频内容。现在视频内容能够通过对等传输网络进行分发，这在很大程度上节约了流媒体服务器的网络流量，并能够在同等带宽的情况下最大限度地提高用户并发数。例如微软媒体服务器协议 MMS（Microsoft Media Server Protocol）属于集中式流媒体服务器技术，而 PPLive、PPStream 等属于对等传输流媒体技术。流媒体技术能够使用户只需下载部分媒体数据就可以持续观看，用户体验优于下载方式。此种方式下的视频文件并不存储在硬盘之中，对用户设备的存储空间要求低，满足 PDA[3]等低存储容量设备的使用。但是流媒体文件不能传送至其他设备或者互联网上。流式传输是建立在多媒体编码技术、网络软硬件技术发展相对成熟的基础上的，如果要保证用户持续观看以达到较好的用户体验，至少要求下载速率高于节目的码率。因此流式传输所要求的技术较下载传输更复杂。

流媒体服务分为直播（LIVE）和点播（VOD, Video-On-Demand）两大类。直播类似于收看电视，用户加入直播系统后就开始收看当前及以后由源发出的节目内容；点播类似于放录像，用户加入点播系统后，不仅可以选择观看什么节目，还可以选择从节目的什么位置观看，执行暂停、恢复、快进、快退、拖动等操作。

2.3 视频传输架构

由于视频服务对于码率、延时以及丢包率的敏感性，视频内容传输在当前的分组交换网络中面临着巨大的挑战。随着互联网视频应用的爆炸性增长，许多架构被提出来以解决视频传输所遇到的问题。目前在视频传输领域流行的架构有客户端/服务器架构、增强型客户端/服务器架构、内容分发网络架构、对等网络架构以及混合式架构等。

2.3.1 客户端/服务器

客户端/服务器是最传统的视频传输架构。客户端直接连接到一台中央 Web/FTP 服务器下载视频文件，或者链接到一台流媒体服务器之上获取视频流。其结构如图 1 所示。

2.3.2 增强型客户端/服务器

与传统客户端/服务器架构只依靠一台视频服务器不同，增强型客户端/服务器架构下客户端从多台部署在互联网上的视频服务器同时获取视频内容，如图 2 所示。由于部署了多台服务器，这种视频传输架构能够有效避免单点故障，并且可以提供多台服务器之间的负载均衡。客户端也能根据当前网络状况、不同服务器的延时等信息动态选择所连接的服务器，以

[3] Personal Digital Assistant，个人数字助理

获得更高的服务质量。

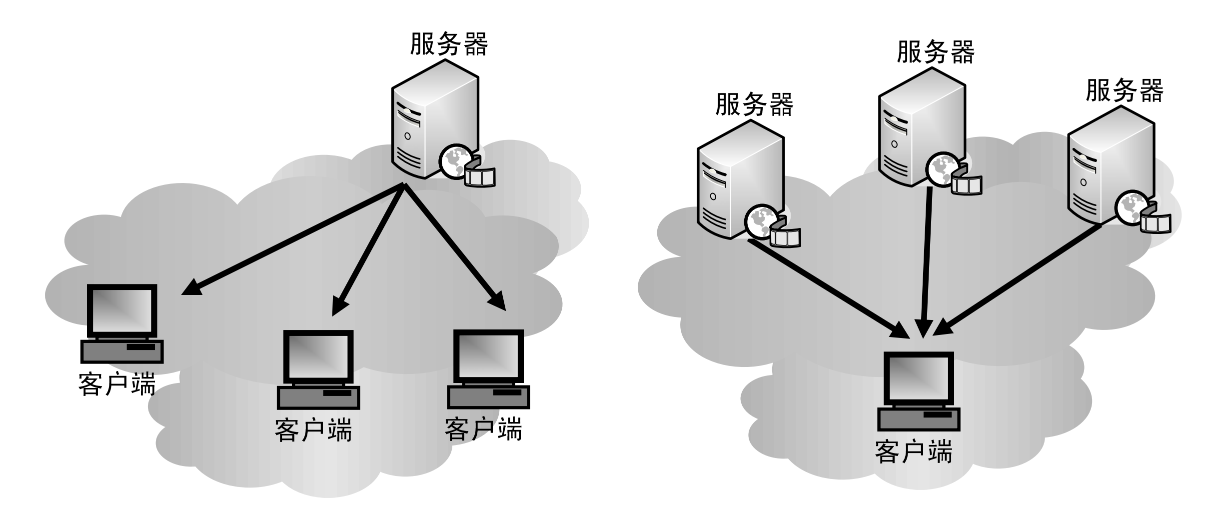


图1. 客户端/服务器架构示意图

图2. 增强型客户/服务器架构示意图

2.3.3 内容分发网络

内容分发网络是建立在现有 IP 网络基础结构之上的一种增值网络，部署在应用层。在内容分发网络架构中，一组内容分发网络服务器被部署到网络边缘，使得用户可以就近获取所需的视频内容，从而缓解网络拥堵，降低客户端的等待延时。其结构如图 3 所示。内容分发网络技术有效地提高了带宽资源的利用率，使流媒体应用在互联网上得到普及。

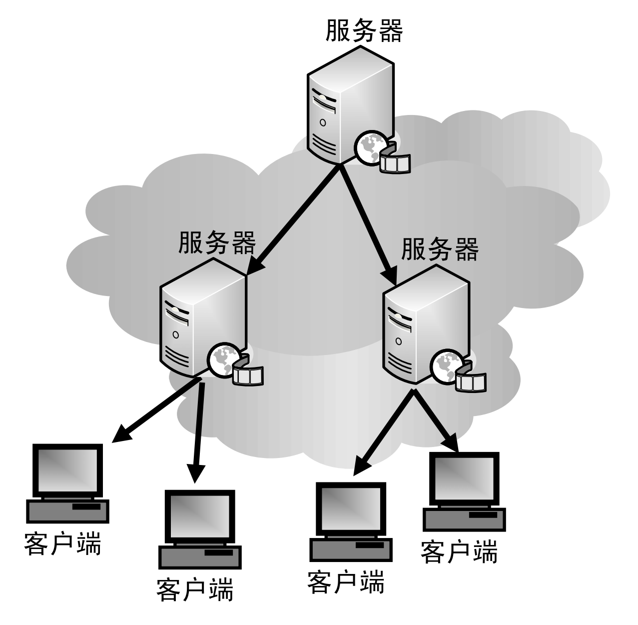


图3. 内容分发网络架构示意图

传统的内容分发网络技术仍然可以看作是一种客户端/服务器架构。尽管传统内容分发网络将服务器能力和视频内容在网络上进行了分布部署，可以在一定程度上提高流媒体服务质量，但其服务的扩展能力取决于内容分发网络服务器节点的部署情况，使其服务能力扩展的成本非常高。同时，内容分发网络架构缺乏弹性的动态扩展能力，难以从根本上提升整个系统的效率。

2.3.4 对等传输

对等网络作为网络体系结构领域内的一项革命性技术，在诸多应用领域在性能、可扩展性等方面具有传统中心式服务无法比拟的优势。目前，对等网络服务得到了大范围的部署，其流量占据了当前互联网流量的主要部分。

对等传输网络中的资源和服务分散在所有节点上，信息的传输和服务的实现都直接在节点之间进行，无需服务器等中间环节的介入，避免了可能的瓶颈。对等传输非中心化的基本特点，使其在可扩展性、健壮性等方面具有优势；通过利用网络中的大量空闲资源，可以用更低的成本提供更高的计算和存储能力。在对等传输中，所有参与者都可以提供中继转发

的功能，因而大大提高了匿名通讯的灵活性和可靠性，能够为用户提供更好的隐私保护；同时，对等传输网络环境下由于每个节点既是服务器又是客户机，降低了对传统客户端/服务器结构中服务器计算能力、存储能力的要求，而且因为资源分布在多个节点，更好地实现了整个网络的负载均衡。

由于以上诸多特点，目前对等传输架构被广泛应用于网络视频传输，例如 CoolStreaming[8]，PPLive[3]，PPStream[5] 等。

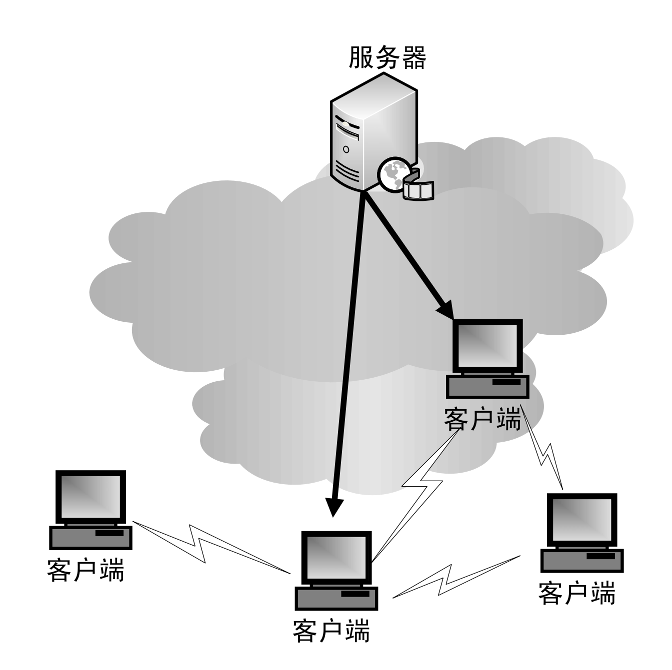


图4. 对等传输架构示意图

2.3.5 混合式

混合式架构融合了以上多种架构模式，以将不同架构的优点集于一身。目前较为流行的是内容分发网络和对等传输技术的融合，例如 Kontiki 公司所提供的混合式视频传输解决方案。

目前内容分发网络和对等传输的融合技术，可以分为以下两种：

1. 对传统内容分发网络架构进行改造，使其服务器节点之间使用对等传输技术来进行内容的分发、交换，并互相备份，以提高中心服务器向内容分发网络服务器进行内容分发的效率。

2. 在传统对等传输网络中加入内容分发网络的管理机制和服务能力，形成一个以内容分发网络为中心，对等传输为边缘的视频传输架构。在引入内容分发网络之后，能使原对等传输网络具有更强的内容及用户的可管理性，同时使整个网络的流量更加有序。

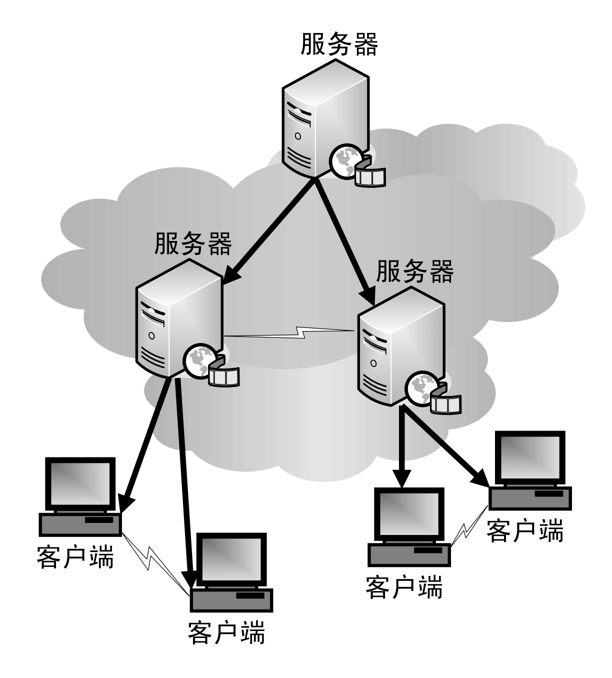


图5. 混合式架构示意图

3 CoolFish 系统结构

3.1 功能

CoolFish 是一个基于对等网络流媒体技术的集视频点播、直播和本地资源播放于一体的软件系统。

用户在下载 CoolFish 客户端后，能够选择我们及时发布的最新影视资源，浏览当前热

门资源，以及搜索所需的视频资源。用户还能通过我们的客户端使用已有的 BitTorrent 种子进行文件下载，并且能够自行发布视频资源。在服务器端，有专门的日志服务器，以及嵌入了 Google Analytics 的 Web 服务器，对用户的行为进行统计及分析。

3.2 系统部署

CoolFish 系统是第一个部署于中国科技网（中国四大 ISP 之一[4]）的对等传输流媒体系统，目前用户主要集中在中国科技网和教育网。自 2008 年 10 月部署以来，大约有 330 万用户访问了我们的系统，日访问人数已经超过 7000 人，同时在线人数超过 700 人。图 6 显示了 CoolFish 系统中用户分布情况。CoolFish 能够支持高达 2.5Mbps 的高清电影，影片的平均码率为 700Kbps，这比当前大多数对等传输流媒体系统的 450Kbps[2,3,4]高出 50%以上。

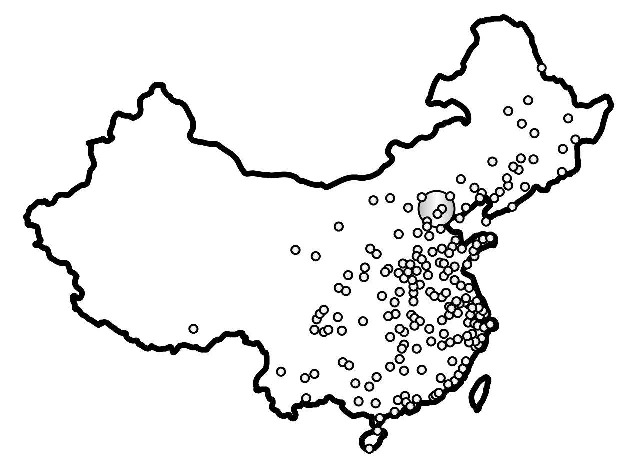


图6. CoolFish 系统中用户分布情况（来自 Google Analytics）

在 CoolFish 系统中，服务器端部署了一台追踪服务器（Tracker），一台节目服务器，一台种子服务器，一台 Web 服务器，以及两台超级节点及一台直播资源发布节点。

1. 追踪服务器

与传统 BT 系统一样，追踪服务器用于收集和提供节点列表。

2. 节目服务器（日志服务器）

节目服务器用于收集各客户端的节目信息，客户端会定期与节目服务器建立连接，将自己观看的点播和直播节目列表上传。节目服务器统计每个节目的下载者数量和发布者数量，并将这些消息存放于数据库中，如果发现客户端下载或发布了新的任务，则要求客户端上传种子文件并保存于本地。

3. 种子服务器

种子服务器接收节目服务器保存的种子文件，并为客户端提供种子下载服务。

4. 网页服务器（数据库）

网页服务器将数据库中记录的节目统计信息和热门节目的排行显示于 CoolFish 官方网站上，并提供节目检索功能。客户端通过其网页服务器通信模块从 CoolFish 官方网站上获取节目列表显示于客户端上。

跟踪服务器
种子服务器
客户端
节目服务器
种子文件
网页服务器
种子文件

图7. CoolFish 系统部署

5. 超级节点（点播资源发布节点）

[4] 中国四大互联网服务商（ISP）:中国电信，中国网通，中国教育网，中国科技网。

点播资源发布节点事实上就是普通的 CoolFish 客户端，管理员使用客户端的发布功能，定期更新发布视频资源。超级节点相当于 BitTorrent 网络中的“做种”用户。

6. 直播资源发布者

直播发布服务器用于将 MMS 引入对等网络，发布直播节目。它与 CoolFish 客户端使用的是同样的直播模块，但是出于直播节目源可控的考虑，目前系统不为普通的 CoolFish 客户端用户提供发布直播节目的功能。

3.3 模块结构

CoolFish 系统分为客户端和服务器两个部分，客户端供一般用户安装使用，用于加入对等网络， 观看点播和直播节目， 服务器供全局使用， 为用户呈现节目、并统计系统信息。

作为对等网络应用程序，客户端是系统的重点，其结构如下：

1. 阿帕奇可移植运行库

阿帕奇可移植运行库（APR Libraries，Apache Portable Runtime Libraries）[9]是阿帕奇（Apache）组织[10]开发的一套跨平台的函数库，它封装了 Windows、Linux、OS/2 等操作系统的应用程序接口（包括进程线程控制、文件系统操作、网络套接口等）。接口提供 C 语言格式支持，通过 APR 库进行系统调用的程序可以在 APR 库支持的所有系统内做到源代码级兼容。早期的阿帕奇服务器程序针对多个平台在程序内部处理不同平台的细节差异，针对不同平台实现不同函数。随着进一步的开发，阿帕奇组织将这些函数单独整理出来，组成阿帕奇可移植运行库，并已成功应用于多个开源和商业项目中。CoolFish 系统将除用户界面和播放器以外的部分建立在 APR 库的基础上，以求将来能方便移植到 Linux 等其它操作系统中（目前发布的 CoolFish 客户端程序仅有 Windows 版）。

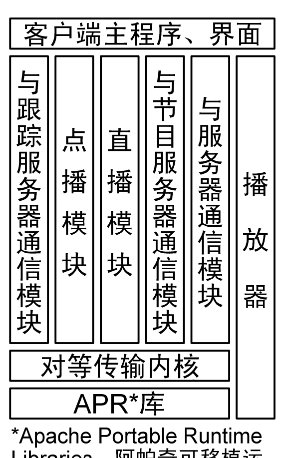


图8. CoolFish 系统模块结构

2. 对等传输内核模块

对等传输内核（P2P Core）模块通过面向对象的方法为上层的各模块定义了网络数据的交换模式。其中比较重要的类为 `Connection`、`ConnectionPool`、`Engine`、`Protocol`。上层各模块需要继承这四个类，并重写其中的虚函数。其中，`Protocol` 代表这个模块的整体，外界与模块的交互都通过这个模块的方法实现；`Connection` 代表这个模块建立的 TCP 连接，例如点播模块定义的 `Connection` 子类表示各节点间用于交换数据分片的连接；模块所有的 `Connection` 放置于此模块的 `ConnectionPool` 中集中管理；`Engine` 负责管理三个线程：`Read`、`Write` 和 `Monitor`。前两个用于负责本模块 `ConnectionPool` 中所有 `Connection` 的读和写，而 `Monitor` 负责管理各 `Connection`，例如在点播模块中，需要定期地统计各个连接的下载速度等参数，移除已经断开的连接等等。因此上层各模块的编写都按照对等传输内核规定的模式，只需实现对读写内容的处理和对各连接的管理即可。这样的设计简化了各模块的开发，为各模块提供了统一的模式，方便了对系统的扩展。

3. 与追踪服务器通信模块

与追踪服务器通信模块用于与追踪服务器通信并获取节点列表。追踪服务器是 BitTorrent 协议引入的集中式服务器，节点上线后需向其下载任务指定的追踪服务器登记，

告知自己是这个任务的参与者，并从追踪服务器获取一些其它参与者的列表以便与它们建立连接。由于 CoolFish 的点播与 BT 协议兼容，自然需要与追踪服务器通信。另外，直播模块的节点列表获取也使用与此相同的方式。这个模块的功能是，调用者为模块制定任务，提供点播节目的 BT 种子或直播节目的哈希（Hash）码，模块将定期向追踪服务器发起连接，获得节点列表，供调用者使用。

4. 与节目服务器通信模块

与节目服务器通信模块负责将本节点下载或观看的点播、直播节目的名字告知 CoolFish 系统的节目服务器。

5. 与网页服务器通信模块

与网页服务器通信模块负责定期与 CoolFish 官方网站建立连接，获取节目列表等信息，呈现于客户端上。

6. 点播模块

点播模块由两个子模块——数据传输和文件管理构成。数据传输模块负责与其它节点建立连接，上传和下载数据分片，提交给文件管理模块。文件管理模块接收数据传输模块下载到的数据，将它们写到存储于本地硬盘的媒体文件中，并在数据传输模块需要某个数据分片时提交给它。另一方面，播放器采取与文件管理模块共享的方式读取媒体文件， 所以文件管理模块要经常接收客户端主程序的查询，告知某段数据是否已完整写入媒体文件可以交给播放器播放，而文件管理模块获知播放位置后应告知数据传输模块，以根据播放位置安排所需分片的下载任务。

点播模块的协议与 BitTorrent 兼容，以便利用 CoolFish 系统外资源。因此用户可以用 CoolFish 打开从其他 BT 网站上下载的种子文件，加入下载网络。如果种子比较热门，可以达到比较好的点播效果。但是，传统的 BitTorrent 协议无法直接应用于流媒体点播。一是 BitTorrent 使用最稀有分片优先下载的策略，虽然对于文件共享来说有利于保存稀有的分片维持整个文件的完整，但分片到达顺序是随机的，难以很快地获得一块连续的数据以供播放；二是 BitTorrent 使用 tit-for-tat[5]策略以鼓励用户为系统上传数据。但所引入的较为频繁的阻塞加大了对等网络的动态性，使得流媒体的稳定性要求难以满足；三是 BitTorrent 协议并未对数据分片的调度和下载任务的分配做出详细的规定，需要依据点播系统的需要，设计较为快速的分片调度方案。

7. 直播模块

直播模块的功能是使节点加入直播对等网络，交换媒体数据分片，发布和观看节目。

作为直播节目的发布端，节点需要连接一个 MMS 服务器，将直播数据流引入对等网络直播系统。这是一种服务器/客户端的方式，要求连接有比较良好的性能（通常 MMS 服务器与直播发布端位于同一台主机）。这是直播网络中数据的最初始来源。本系统直播模块的设计允许多个直播发布端连接至一个 MMS 服务。由于它们根据 MMS 服务计算得到的哈希码是相同的，因此实际上是组成了一个统一的直播网络，而不是作为各自直播网络的唯一的发布端。这样的好处是组成了较大规模的直播网络，有助于发挥对等网络的优势，并可以对

[5] 一种博弈策略。即首次以善意回应对方，然后根据对方回应是否友好立即以对等的方式回应，见 http://en.wikipedia.org/wiki/BitTorrent 。

同一个 MMS 服务进行多处发布，分散单一发布端的压力，且不会造成不必要的重复。

作为直播节目的观看端，直播模块除完成数据分片的下载和传输外，还在本地开启一个 MMS 服务，下载到的数据分片被顺序提交到本地 MMS 服务器，播放器直接连接这个 MMS 服务器端口观看节目。这样，客户端主程序与直播模块的交互非常简单，只需告知要观看什么节目，并让播放器连接本地 MMS 服务器即可。

4 CoolFish 使用的算法

4.1 类 BT 协议

4.1.1 BT 协议介绍

BT 是 BitTorrent 的缩写，是一种分发文件的协议。BitTorrent 目前已经成为 互联网上最为流行的对等传输协作文件共享分布系统。BitTorrent 在被组织起来的系统节点中共享相同的文件，并将此共享文件快速、有效地分布和复制到系统中所有其它节点上。BT 利用每个节点的带宽将共享文件有效地复制到大量需求此共享文件的节点上，目标是能够快速有效地分布和下载大尺寸的共享文件。在 BitTorrent 系统中，一个基本的概念是将一个共享文件分成很多等尺寸的文件块（一般为 256K）。一个系统参与节点可以并行地从多个不同的节点下载不同的文件块，并且同时上载自己拥有的文件块给其它需求节点。BitTorrent 系统采用哈希（SHA1）的机制来保证下载来的所有块可以顺利重组。另外在 BitTorrent 系统中，节点之间的数据块交换是通过 tit-for-tat 策略来激励和推动的。

下面对 BitTorrent 协议中所使用到的一些关键算法进行介绍。

1. 阻塞与尝试性疏通算法

在 BitTorrent 中使用阻塞算法有多重原因。其中一个原因是该算法能够在节点间实现 tit-for-tat 激励机制，从而确保稳定的下载速率，提高用户之间共享机率。

BitTorrent 系统的每个节点一直与固定数量的其它节点保持疏通（unchoke）。在 tit-for-tat 激励机制下，每个节点上传数据块给自己的邻居节点，节点将严格地根据当前的下载速率来决定哪些节点应该保持疏通，哪些节点应该被其拒绝上传服务，从而确定它的连接节点集合。拒绝上传服务被称为阻塞（choking），相应的算法被称为阻塞算法。

BitTorrent 系统同时也采用了尝试性疏通方法去选择节点上传。在尝试性疏通方法中，每个参与节点除了根据 tit-for-tat 策略选择固定数量节点去上传之外，还会随机选择一个请求下载的节点去上传数据，但是它并不会考虑这个节点过去的上传速率情况。也就是在任何时候，每个节点都拥有一个称为尝试性疏通的连接，这个连接总是保持疏通状态，而不管它的下载速率是怎样。增加尝试性疏通主要出于两个目的:其一是节点可以尝试连接一下未连接过的节点，找出一个可能比现有连接更好的连接；其二是当新节点刚加入系统时没有任何文件数据贡献，尝试性疏通方法可以保证新加入节点快速下载到部分文件内容，从而可以快速参与到系统中与其它节点交换数据。尝试性疏通每隔 30 秒循环转换被尝试性疏通的节点。30 秒足以让上传能力达到最大，下载能力也相应的达到最大。

2. 最稀有数据块优先

最稀有数据块优先（Rarest First）是指一个下载节点在选择下载的目标数据块时，首先选择其所有邻居节点中该数据块拥有者的数目最少的那个数据块。也就是最稀缺的那个数据

块来下载。这个策略被称为“最稀有数据块优先策略”。在这个策略下，整个系统会趋向于一种优化的状态，数据块会在系统中均匀分布，提高整个系统的性能。并避免了拥有稀有数据块的节点过早离开系统后造成的抖动，提高了系统的稳定性。

4.1.2 CoolFish 对 BT 协议的修改

CoolFish 点播模块中，我们对传统 BitTorrent 协议做了一些修改，使得 CoolFish 客户端既能与一般 BT 客户端交换数据， 又能更好地满足点播的需要。首先，我们以“最需要分片优先下载”取代“最稀有分片优先下载”；其次，我们使用十字链表管理下载分片的任务。第三，在对上传数据的管理方面，为本节点的各个邻居 P_j 引入优先级，优先响应优先级高的节点的上传分片请求。依据速度改变优先级的方法事实上相似于 BT 的 tit-for-tat 策略，优先为向本节点提供高质量服务的节点服务，鼓励节点为系统贡献传输能力，但是采用了比 tit-for-tat 的“阻塞—非阻塞”更加平滑的判决方式，更适应流媒体点播对数据流稳定的需要。

4.2 多任务下载

普通视频点播系统，用户只能下载及分享当前正在收看的节目内容，我们称之为单任务缓存策略(STC)。由于 CoolFish 系统完全兼容 BT 协议，所以提供了类似于普通 BT 客户端的多任务下载技术(MTC)，用户在收看当前节目的同时，能够下载及分享其他节目内容。此技术在一定程度上提高了视频的分享率及上传速度。如图 9 所示。

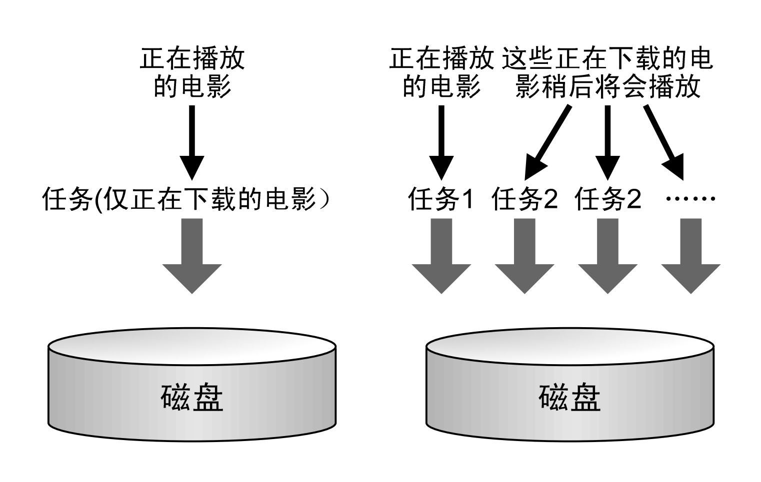


图9. 单任务缓存与多任务缓存

4.2.1 Mediacoop

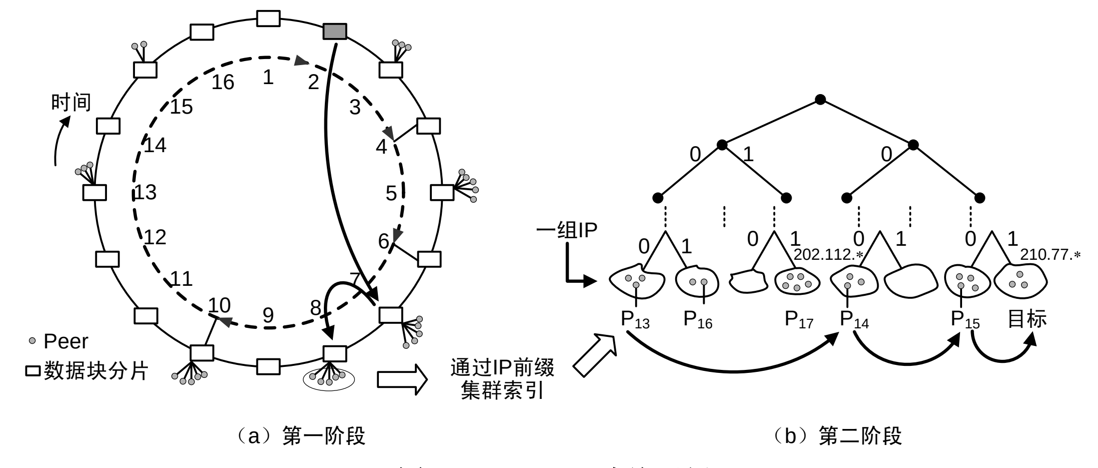


图10. Mediacoop 查询示例

Mediacoop 是一种结合了内容及质量匹配的对等传输视频点播结构化查询算法。Mediacoop 通过播放点距离以及全局网络延时来提供查询服务，以降低客户端的启动延时及跳转延时，并提高播放的连续性。Mediacoop 的查询过程可分为两步。首先是内容匹配查询，

根据路由表中邻居的播放点信息，通过递归查询，找到一组含有所需内容的附近播放点作为节点，如图 10（a）。其次，根据全局网络延时表，选择延时小的节点进行连接，如图 10（b）。

实验显示 Mediacoop 算法能够有效降低客户端的启动延时以及跳转延时，如图 11。

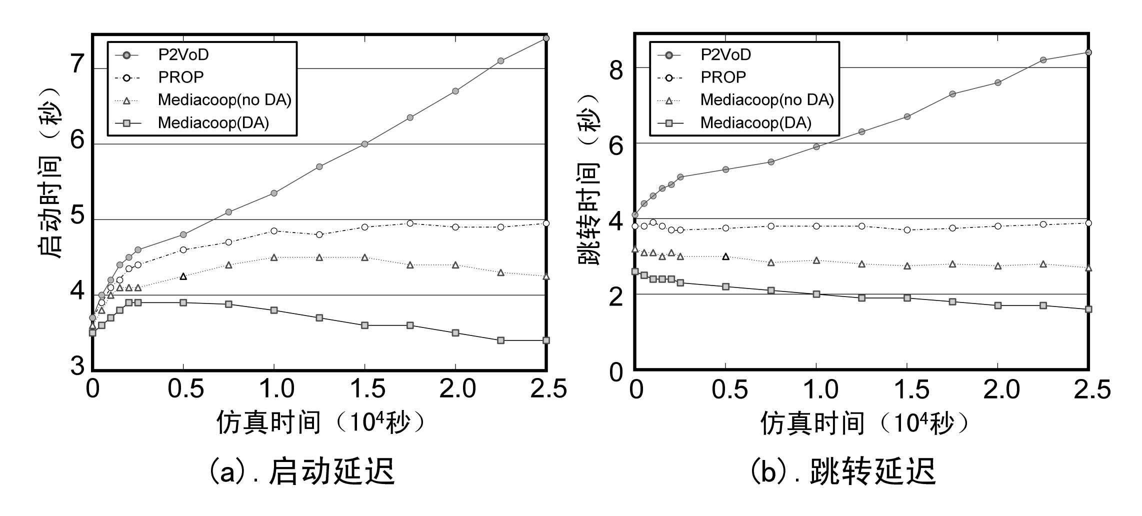


图11. 启动延时以及跳转延时

4.3 直播数据交换策略

目前的 CoolFish 直播模块的数据交换策略使用经典的“拉”策略，节点定期地发送缓冲区中数据分片拥有情况（即 PieceMap），缓冲区被分割成若干个窗口，窗口是数据提交给本地 MMS 服务器的单位。节点依据所需分片在邻居中的分布情况和各邻居的能力分配下载任务。 对于每个分片的请求， 估计一个等待时间。如果超过等待时间后分片还未到来，则启动“一次超时机制”，向拥有此分片的其他各邻居请求这个分片。如果还没有按时收到分片，则表明本节点需要的数据已大大落后于网络中各节点的播放进度以致数据失效，或者出现了邻居的大量断线。这时需启动“二次超时机制”，视需要重新与邻居建立连接，并依据邻居 PieceMap 重新确定本节点的缓冲区起始位置。在下载分片的过程中，如果最前面的窗口已经填充完整，则可将它提交给本地 MMS 服务器，并将缓冲区整体前移。有时候，最前面的窗口可能只有个别分片未下载完成，如果继续等待或启用“一次超时机制”，会减慢缓冲区整体向前平移的速度，很容易落后于系统中其它节点以致进入“二次超时”的状态，而且播放器可以很好地应对个别分片丢失的情况（也可用空数据分片替代缺失分片），因此可以直接将此窗口提交给本地 MMS 服务器，并将缓冲区前移，不因个别分片影响整个下载进程。

5 总结

近年来，对等网络的应用日益广泛，研究不断深入。流媒体是对等网络得到重要应用的一个领域。对于对等网络流媒体的研究已经历了多年的发展。对等网络流媒体技术充分利用了各参与节点相互协助的特性，取得了很大的进步，并出现了诸多较为成功的商业系统。CoolFish 是一个基于对等网络流媒体技术的集视频点播、直播和本地资源播放于一体的软件系统。目前系统上线服务不久，还有一些有意义的工作留待将来完成。例如依据对用户行为的分析提供个性化的服务，加入分布式检索的功能，以及实现多种子多协议的互通以进一步提高下载速度等。

（下转第 9 页）